A Biologically Motivated Computational Architecture Inspired in the Human Immunological System to Quantify Abnormal Behaviors to Detect Presence of Intruders.

Omar U. Flórez-Choque[1] and Ernesto Cuadros-Vargas[23]

[1] Computer Science Department, National University of San Agustín. Arequipa, Perú. omarflorez19@gmail.com
[2] San Pablo Catholic University
[3] Peruvian Computer Society ecuadros@spc.org.pe

Abstract. In this article is presented a detection model of intruders by using an architecture based in agents that imitates the principal aspects of the Immunological System, such as detection and elimination of antigens in the human body. This model is based on the hypothesis of an intruder which is a strange element in the system, whereby can exist mechanisms able to detect their presence. We will use recognizer agents of intruders (*Lymphocytes-B*) for such goal and macrophage agents (*Lymphocytes-T*) for alerting and reacting actions.
The core of the system is based in *recognizing abnormal patterns of conduct* by agents (*Lymphocytes-B*), which will recognize anomalies in the behavior of the user, through a catalogue of Metrics that will allow us quantify the conduct of the user according to measures of behaviors and then we will apply *Statistic* and *Data Minig* technics to classify the conducts of the user in intruder or normal behavior. Our experiments suggest that both methods are complementary for this purpose. This approach was very flexible and customized in the practice for the needs of any particular system.

1 Introduction

Although the passwords, iris and retina readers as well as the digital signatures work well, a serious problem exists when these controls are overcome by stealing of the password, modification of the firmware or by stealing of the user's smart card. For intruders that masqueraded as valid users enter in the system and they carry out diverse actions that put in risk the integrity of the system [1]. Then, it arises the need of detecting those intruders, by knowing they have a different behavior pattern from the true user, with the result that it firstly is necessary to define a mechanism that allows us to measure each behavior of the user, so when comparing behaviors we will find a numeric value that allows to differ them.

In that sense it is admirable the way like the Immunologic System works. The Immunologic System is an important defensive system that has evolved in the vertebrate beings to protect them of microorganisms invaders (bacterias, virus, so on). In the moment when a wound appears, the white globules detect an antigen (intruder) in the human body through the sanguine torrent. Then appear two types of Lymphocytes among other agents, the *Lymphocyte-B* and the *Lymphocyte-T*. The Lymphocytes-B recognizes the antigen through proteins of complement (18 proteins that exist in the plasm and that are activated in a sequential way) and then these Lymphocytes produce Antibodies that can be able to face the identified intruding agent. The Lymphocyte-T is responsible for reactive functions destroying the strange substance, in view of each antibody is specific for each microorganism, the reaction of the Lymphocyte-T will vary according to the antigen recognized by the Lymphocytes-B.

It is remarkable the adaptability and the persistence of the information in the human body, since the white globules remember biochemically the analyzed antigen, so future answers will be quicker and more exact.

It is also interesting, and is the aim of this paper, to present the foundations of this mechanism of *detection and defense against intruders* toward a computational system.

The rest of this paper is organized as follows. In Section 2 is analyzed the architecture of the presented model by describing the hierarchical relationships between each one of the agents. In the Section 3 the functions of the used agents and their analogy with the Lymphocytes of the Human Immunologic System are described. In the Section 4 we define a catalog of metrics that will allow us to quantify the behavior of the user based on four behaviors: Effort, Memory, Trust and Special Requirements. In Section 5 we discuss the model of detection of intruders based on Statistical and Data Mining methods, which classify the vectors of behaviors in behaviors belonging to either intruders or normal users. In the Section 5 and 6 we discuss the obtained results. Section 7 briefly describes related works and specifies our contribution. And lastly, section 8 provides some conclusions.

2 Architecture of the model

The architecture is based on agents with different roles: Control-Reaction, Maintenance and Net, the model also includes databases that are organized in a distributed way.

According to the Figure 1, the *Control-Reaction Agents* are distinguished. They proactively read information of activity, find patterns and trigger alarms (through Lymphocytes-B) and they perform protection actions (through Lymphocytes-T). Besides, they control the use of resources wasted by the users, monitoring all their actions, in such a way this type of agents identifies profiles based on the account, resource and action that the user carried out. This type of agents are Lymphocytes-B and Lymphocytes-T.

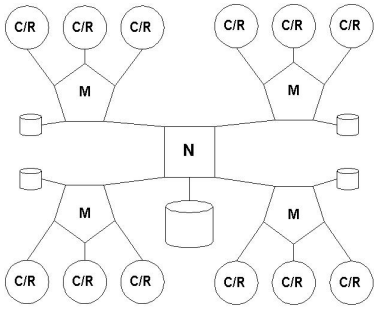

Fig. 1. Architecture of the Model based on agents

Then, we have the *Maintenance agents*, which create or delete Control-Reaction agents, and elimate redundant data, besides compression and local encoding of data is performed by this agent. This agent also maintains the database, where the profiles of activity and accounts of the users are stored, receiving the queries formulated by the Control-Reaction agents, execute them and return the results so that the Control-Reaction agents perform the necessary actions. There is one agent of this type for each authentication server because existis a single database inside this type of servers in the system. Lastly, the *Net Agent* creates or deletes Maintenance Agents. In view of this agent has a whole vision of network. It can detect other types of attacks, such as multi-host attack or *Denial for Service* (DOS), and besides it can realize the filter of packages on the net.

We use agents and not neural nets or other technology due to the heterogeneity presented in the problem: The *Lymphocytes-B* agents should learn that actions of the users change in the time and they should adapt their profiles according to these changes in view of each *Function of Behavior* is individual to each user. The *Lymphocytes-T* agents should learn how to trigger different security policies (resource denial, account elimination, lockout of account, restricted access) according to stored patterns of behavior. And the *Net agents* should learn how to recognize abnormal patterns of activity based on present information in the whole net to recognize multi-host distributed attacks . Therefore, when we use agents, we combine the necessity to use three great actors in the system.

3 Components of the architecture

3.1 Lymphocyte-B

These agents have the task of monitoring the actions of the user, identifying of this way the profiles, which store each action that the user carries out at one time in a certain resource. This information allows us to measure possible behavior changes in the account of the user and the later detection of an antigen

in the system. It is possible to measure the behaviors of an user with metrics, which identify the behavior of the user in the system, according to this method a group of behaviors will define in a unique way the behavior of the user. Then, once obtained the vectors of user behaviors that began session, it is possible to compare the values of those vectors with the *Function of Behavior* of the user, and to foresee if there are abrupt variations in the behavior, which would reveal us an atypical and suspicious behavior, therefore the system will react.

The Lymphocyte-B agent has the particularity of requesting queries to the database that corresponds to it (there is one database for each authentication server), for example this agent requests information, creation, upgrading and elimination of profiles, but not carrying out them, for this, the Lymphocyte-B agent sends messages to the maintenance agents (wich are in charge of maintaining the database) and receives messages from agent of maintenance with the datasets of the query performed. In this type of agents, this information is necessary to be able to differ the behavior value calculated on relation to values of previous behaviors. For example, if the user gMoore@cisco.com usually uses its account to read information on internet, and then, in other session, the Lymphocyte-B agent monitors activities with a high usage of CPU due to compilation activities. It will compare this value with previous behaviors by formulating queries that will be executed in the database that stores the profiles and it will detect that this behavior changes is not normal, then the Lymphocyte-B will send a message to the corresponding agent so that it reacts because of this anomaly.

This mechanism resembles the recognition of a strange agent in the human body by the immunological system.

3.2 Lymphocyte-T

This type of agent has the task of reacting when an anomaly appears. Once detected the anomaly by the Lymphocyte-B agent, the Lymphocytes-T agent can give a message of alert, to expel the user, lockout the account, refuse an action to the user or ignore it depending on the case.

This mechanism resembles the reaction of the human body when a strange agent arises.

3.3 Maintenance Agent

This agent is the only agent allowed to manipulate the database, which stores the behavior information of each user. In fact, this agent executes the queries received from *Lymphocyte-B* and *Lymphocyte-T* agents to carry out them and returns datasets with the result of the query to the agent that requested it. If all the agents manipulated this database, the information would be outdated, and not synchronized, besides the traffic of net would be considerably increased since the use of the cache unit would be null. There is one *Maintenance* agent for each authentication server.

3.4 Net Agent

On the other hand, Mauro [2] filters all the packages of the net, so like a sniffer to read the headers of these packages and to see the executed command and starting from that to formulate the possible behavior of the user. This method has the advantage of not overloading the net considerably, however the presence of techniques of encryption could not make it appropiate, anyway, if this mechanism was implemented, this agent would be whom to implement it.

It is possible to take advantage of the geographic distribution of the system to achieve intrusion tolerance using the *fragmentation-redundancy-scattering* technique [3] by cutting all user sensitive data into fragments which are encrypted, stored and replicated among all the databases in the authentication servers. A high level of granularity in the data is obtained in view of several fragments together are not enough to disclose the information of the user. In fact, each *Lymphocyte-B* agent take a local decision to reject an intruder according changes on behavior which are locally stored, then this local decision is broadcasted to the other *Lymphocytes-B* agents and all the decisions, included the local decision, are locally voted and the rejection is locally trigged or not. This technique is called *majority voting* and ensures that false alarms can not be trigged.

4 The Vector of Behavior

If we want to transfer the mechanism of recognition of antigens, carried out by the Lymphocyte-B, by means of complement proteins toward a computational system we will need another mechanism that allows us to differ the intruders quantitatively since an user is a strange agent. In this article we propose a catalog of metrics that enables us to compare different user accounts independently of the operating system, programming language or implementation done, because they are based on changes of their behaviors.

Four behaviors of user can be identified in this discussion: *Effort*, *Memory*, *Trust* and *Special Requirements*. Each behavior reflects a great part of the way of behaving of the user into the system, for instance there will be user with great amount of work, user with low capacity of memory and users with special requierements like low display resolutions. Each aspects reflect a behavior of the user. This behavior is dynamic because it changea in the time. Therefore the total user behavior would be composed by a vector of behaviors, where each dimension of the vector is associated with a specific behavior.

Once we have all the dimensions from calculated behaviors, the value of behavior vector can be represented by a measure of distance to quantify the divergence among behaviors. Well known distances are Euclidean, Euclidean normalized, metric of Tchebycheff, Mahalanobis, and Tonimoto. We choose Euclidean distance because it exhibits some very interesting properties: it is variant to scale change and it depends on the relationships among the variables.

4.1 Behaviors

Effort This behavior reflects the quantity of work performed by an user, for example a high value of this behavior would mean that the user is using too much CPU time, maybe compiling a program, which in the worst case can be a *Troyan Horse* or a *port scanner*. Besides the user can be writing too much data on the hard disk, in the worst case it can reveal the presence of a virus in the System, on the other hand, if the user is producing plenty net traffic, it can reveal a typical DoS attack. Therefore, these aspects identify the quantity of effort of an user in the system. We show four metrics, which allow us to quantify this behavior:

1. $consumptionCPU = \frac{CPU\ time}{session}$
2. $readWriteDisk = \frac{Kbytes\ R/W\ in\ disk}{session}$
3. $trafficNet = \frac{Kbytes\ of\ data\ transferred\ in\ the\ net}{session}$
4. $durationSesion = Duration\ of\ the\ session$

Applying the *Euclidean distance*, the coefficients of the behavior of Effort are defined as:

$$\text{Effort} = \sqrt{\text{consumptionCPU}^2 + \text{readWriteDisk}^2 + \text{trafficNet}^2 + \text{durationSesion}^2} \qquad (1)$$

A considerable variation in the value of this behavior would involve that an user carries out activities that before he did not make which can be due to changes of departments, promotions in the work or the presence of an intruder masqueraded in the account. The system would detect this as an abnormal behavior for the user. A high value of this coefficient will also mean a great quantity of work carried out in the account of the user.

Memory This behavior reflects the amount of mistakes of the user due to the forgetfulness that he can experience, for instance the forgetfulness of the password, most of true users remember the password very well and they enter to the account in the first intent, however we should also consider elder users and, worse even, users with dyslexia that are not able to remember with easiness a password, in any way, this grade of forgetfulness defines an aspect of the behavior of the user.

We should also consider that when an intruder enters to the system, this intruder tries firstly to obtain information from the account by means of commands of information of the system [1], again, an average user will not need too much information about itself to begin to work normally. A similar case is when an user often uses a group of commands, while this user uses more this group of commands, less errors will happen when writing them, although we should consider users with low skill in the use of the keyboard and elder users with Parkinson disease [4].

Therefore we present three metric that qualify this behavior:

1. $wrongCommand = \frac{Commands\ written\ incorrectly}{session}$
2. $wrongLogin = \frac{Number\ of\ invalid\ logins}{session}$
3. $commandInformation = \frac{Number\ of\ times\ that\ information\ commands\ are\ executed}{session}$

Applying the *Euclidean distance*, the coefficients of the behavior of Memory are defined as:

$$Memory = \sqrt{wrongCommand^2 + wrongLogin^2 + commandInformation^2} \tag{2}$$

Trust This behavior reflects how reliable is an user in the system. There are users prone to be attacked [1], for example users that elect as password a word that is in the dictionary, without the use of uppercase, numbers or special characters; this makes the user to be not very reliable before a brute force attack. This is a subjective measure and we will say that a password with uppercase, numbers and special characters has a value of 0, an alphanumeric password has a value of 3 and a simple password has a value of 10. Then exist users that for curiosity or with purpose try to read, write or execute files and for obtaining information that does not correspond them, because they do not have the enough privileges to make it. Thereby, we can count the number of invalid accesses to define a feature of the user behavior that indicates if the user is not very reliable. Finally we can try to measure the fact that an user hides information through encryption of data. This last metric is relative, however most of hackers try to *hide their fingerprints* through encryption, so that the administrator of security can not examine the information that stores an account, although this fact is not so serious, however an excessive quantity of encrypted information is very suspicious.

Therefore, we present three metrics to quantify the trust of the system in a certain user:

1. invalidActions = $\frac{Number\ of\ invalid\ actions}{session}$
2. complexPassword = Complexity of password.
3. encryptedInformation = Amount of encrypted information stored in the account of a user.

Applying the *Euclidean distance*, the coefficients of the behavior of the Trust are defined as:

$$Trust = \sqrt{invalidActions^2 + complexPassword^2 + encryptedInformation^2} \tag{3}$$

Special Requirements This behavior reflects special needs that the user requires of the system, for example if an user is always connected by modem and then suddenly carries out a connection by wireless devices, this change of behavior appears suspicious for the system, then we assign a value of 0 if

the connection is carried out on intranet, 3 if it is carried out by modem and 10 if it is carried out by wireless connection. It is also necessary to identify those users that are not authenticated by means of common mechanisms as their password, but through special devices as iris or retina readers, detection of faces, digital certificates, touch sensitive screens and so on. Whereby, if in the authentication process the password is introduced by keyboard, the values of 0 is assigned to this metric. If digital certificates are used, it is assigned a value of 3 and 10 in other cases. However we would keep in mind the possibility that an user changes the type of authentication, among other reasons the user suffers some temporary or permanent disability, for example, blindness which prevents to use iris readers, in this case the administrator would receive many warnings indicating a suspicious change of behavior. Lastly, we will try to quantify the fact that the user uses requirements of accessibility to work normally in the system [4]. Users without superior extremities will have difficulty to use the keyboard, for this is required a virtual keyboard on the screen, on the other hand users with astigmatism, myopia or permanent blindness require a magnificator screen, or a screen reader respectively. Then if an intruder changes these options, clearly it implies a change in the behavior of the user. This way if the user does not use any requirement of accessibility, a values of 0 is assigned to this coefficient, a value of 10 is assigned in other cases.

Then, we present three metrics to try to quantify special necessities of an user:

1. $typeConnection = Type\,of\,connection\,to\,the\,system.$
2. $typeAuthentication = Type\,of\,authentication.$
3. $reqAccessibility = Requirements\,of\,Accessibility.$

Applying the *Euclidean distance*, the coefficients of the behavior of the Trust are defined as:

$$\text{Special Requirements} = \sqrt{\text{typeConnection}^2 + \text{typeAuthentication}^2 + \text{reqAccessibility}^2} \quad (4)$$

5 Experimental results

We estimate our results over 200 registers, each register stores a behavior vector. We will use this method *5-fold cross validation* to estimate accuracy. The crossed validation is the standard method to estimate predictions on test data for Data Mining and Neural Nets [5]. We split the total of registers in 5 groups of same size. We use 4 groups for the training of the model (Training Set) and the remaining one for the evaluation of the model(Test Set), then we repeat the process 5 times leaving-one-out different partition in each cycle as test group. This procedure gives us a very reliable measure of accuracy of the model. Then we average the result of these 5 groups to recognize how the model was executed over the whole data. Then, we will use the *ROC curves* (Receiver Operating

Characteristics) to visualize the accuracy in the classification, the ROC curves are commonly used in the medicine for taking of clinical decisions and in recent years have been increasingly adopted by the communities of investigators of Data Mining and Learning Machines [6].

Given a classifier and one group of instances, there are 4 possible states in which the instance can be classified:

- True Positive (TP).- Intruder that is classified as Intruder by the system.
- True Negative (TN).- Normal User that is classified as Normal User by the system.
- False Positive (FP).- Normal User that is classified as Intruder by the system.
- False Negative (FN).- Intruder that is classified as Normal User by the system.

We are interested in the *rate of Intruders* detected by the model (*True Positive*) and in the rate of *"false alarms"* or Normal Users that are classified as Intruders (*False Positive*). To build the ROC curves we are interested in the following metrics:

- The rate of detected intruders: $\frac{PV}{TP+FN}$
- The rate of false alarms: $\frac{PF}{FP+TN}$
- The global accuracy: $\frac{TP+TN}{TP+TN+FP+FN}$

The results of the experiments are summarized in the Table 1.

Table 1. These are the results of classifying the behaviors of the user in the account gMoore@cisco.com organized by the type of used classifier. In spite of the method of Deviation Standard had the smallest rate of false alarms and the highest rate in Detection of Intruders, the method of Decision Trees detected different registers belong to "*true intruders*" which had not been detected by the Deviation Standard method.

Classifier	Deviation standard	Decision trees
True positive (TP)	77	68
True negative (TN)	73	61
False positive (FP) (PF)	7	14
False negatives (FN)	3	7
% Detection of Intruders	96.25 %	80.00 %
% False alarms	4.75 %	18.67 %
% Total accuracy	93.75 %	80.63 %

6 Discussion

Both considered approaches (Standard deviation, Decision trees) works very well in the detection of intruders, in spite of Decision trees had a lower value

than Standard deviation, it detected different registers from intruders that those detected by Standard Deviation. This suggests that both classifiers can be mixed to define a stable and reliable method to implement intrusion detection schemes.

6.1 Statistic

In the Figure 2, we visualize the accuracy in the detection of intruders through techniques of Statistic and Data Mining. Both ROC curves are concave therefore they have a good exchange between detection and false alarms rates.

The classifier based in techniques of Statistic obtained the highest rate of *detection of intruder* due to this method is based in more recent behaviors of the user. Besides it adjusts by itself in the time in a learning way, based in the tendency of the behavior function in the time. This method also had the highest *global accuracy rate*.

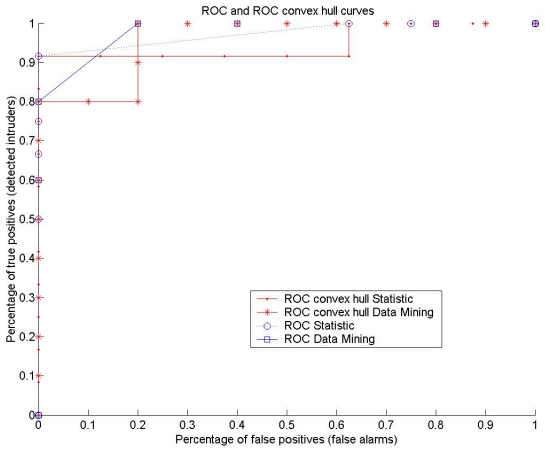

Fig. 2. ROC curves for the models of detection of intruders based on Statistical and Data Mining methods. Notice that curves generated by Deviation Standard have a higher detection of intruders rate regarding the curves generated by Decision Trees, although they intersect in a percentage of 92% in the detection of intruders for a percentage of false alarms of 14%.

6.2 Data Mining

The model of detection of intruder based on decision trees had the higher false alarm rate. In spite of having 80% in the *detection of intruders rate* the difference regarding the model based on Deviation Standard was 14%. The two models intersect in a *false alarms rate* of 14% with a *detection of intruders rate* of up to 90%.

7 Related works

An alternative approach is taken by Forrest et al. [7], who is focused to determinate normal behaviors for privileged process, that is, process that run as root (*sendmail* and *lpr*). However in this model is difficult to detect an intruder is masquerading as another user because of this approach is based in low level features (ports, system calls, processes). Our approach rely on more meaningful features (Effort, Memory, Trust and Special Requirements), which identify more exactly the behavior of the user into the system. These features are inherent to the user, therefore an user can not forget them and a intruder can not guess them. Besides we tried to emulate an architecture inspired in the principal functions of the Human Immulogical System through lymphocytes (B and T).

8 Conclusions

We believe the proposed model provides a base to implement a system capable to recognize intruders according to behaviors of the owner of the account, in that sense, we believe that an intruder can guess the *password* of an user, but difficultly , will be able to guess the *behavior* of the user. We showed that the model based on statistical techniques had the higher detection of intruders rate, 96.25%. Although the model based in techniques of Data Mining had the higher false alarm rate, 18.67%. Therefore we recommend mix both methods, these data can help to decide an security administrator to use one of the models or both, according the specific necessity of security. For example in an critic security environment, is more important to have a high grade of detection of intruders. On the other hand, in mail servers can be more important the availability of the service, in spite of this service do not be so exact, thereby a rate of false alarms of 18.67% can be acceptable.

Acknowledgements

The author wants to thank very especially the teachings and advices from the teachers of the National University of San Agustín at Arequipa Dr. Ernesto Cuadros-Vargas and Dr. Luis Alfaro Casas. Besides, I would like to express my deep gratitude to Dr. Rosa Alarcón Choque, University of Chile at Santiago, for her invaluable encourage.

References

1. K. Mitnick. The Art of Deception. *Wiley.* December, 2002.
2. A. Mauro. Adaptative Intrusion Detection System using Neural Networks. Conference of ACME! Computer Security Labs. November, 2002.

3. Y. Deswarte, L. Blain, and J. C. Fabre. Intrusion tolerance in distributed computing systems. In Proc. Symp. on Research in Security and Privacy, pp. 110-121, Oakland, CA, USA. 1991. IEEE Computer Society Press.
4. S. Burgstahler, Sheryl. Working Together: People with Disabilities and Computer. University of Washington. DO-IT. 2002.
5. R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. IJCAI. 1995.
6. T. Fawcett. ROC graphs : Notes and practical considerations for researchers. Technical report, HP Laboratories, MS 1143, 1501 Page Mill Road, Palo Alto CA 94304, USA. 2004.
7. S. Forrest, S. A. Hofmeyr. A. Somayaji, and T. A. Longstaff. A sense of self for Unix processes. In Proceedings of 1996 IEEE Symposium on Computer Security and Privacy, pp. 120-128 (1996).